IITG.ai Paper Discussion

FaceNet : A Unified Embedding for Face Recognition and Clustering

Vishal Agarwal

Overview

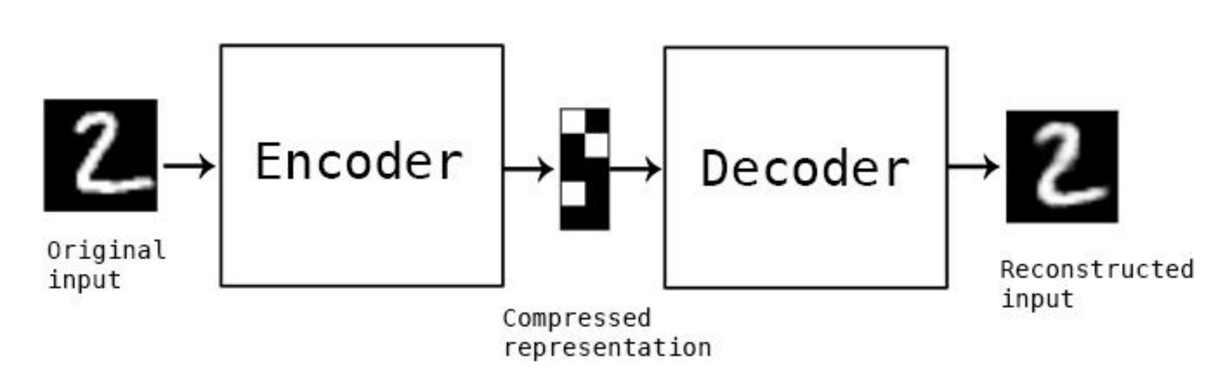
- Learns mapping from face images to an Euclidean space where distance correspond to the measure of similarity.
- Generates a 128-dimensional embedding.
- Embeddings can be used for facial recognition, verification and clustering.
- Introduced triplet network and triplet loss function.
- Idea :
  - Faces of same person : small distance between embedding.
  - Faces of distinct person : large distance.
- Data-driven approach

Application

- Face verification : Is same person?
  - Simple Thresholding
- Face recognition : Who is the person?
  - k-NN Classification.
- Face Clustering : Find common people
  - K-means clustering

Previous work

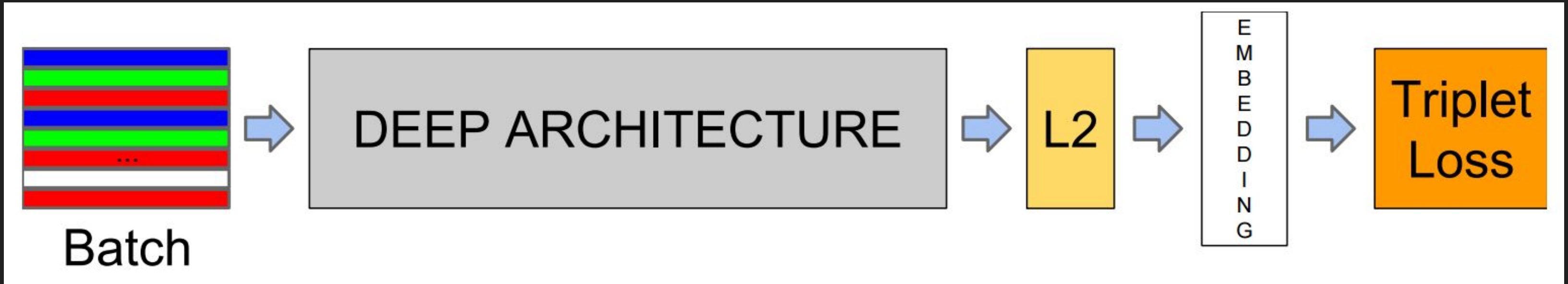
- Most famous architecture for this task : Autoencoders.
- Bottleneck layer used for generating latent representations and then used for facial recognition tasks.
- Also post processing such as model concatenation or SVM classification.

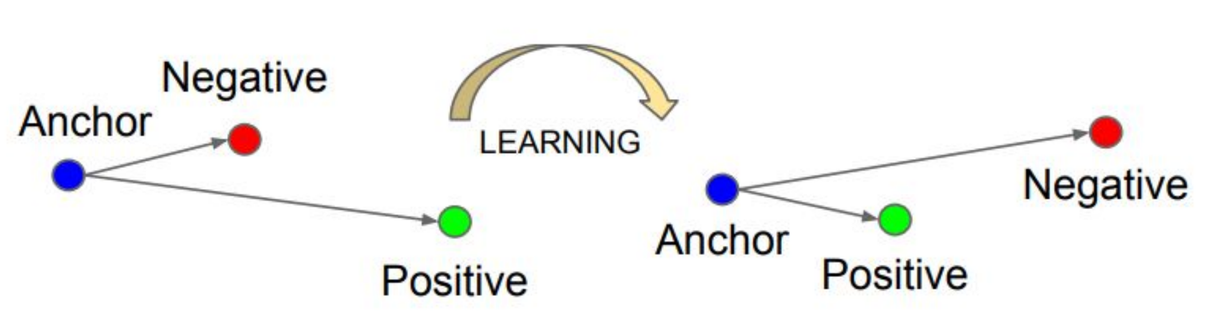


Autoencoder model

(Source:https://blog.keras.io/building-autoencoders-in-keras.html)

Model





FaceNet Model and Triplet Structure

Model

- Generates a 128-dimensional embedding.
- Inputs are given as **Triplet** :
  - Anchor : example under consideration.
  - Positive : example belonging to same class as anchor.
  - Negative : example belonging to different class as anchor.
- Triplet Loss function introduced.
- 2 different CNN architectures used
  - Zeiler & Fergus Net
  - Inception (GoogLeNet)
- For similar performance, the latter reduces the number of parameter by 20x and number of FLOPs for computation by 5x.

Model

- Objective : Learn a mapping function f(x) such that the L2 distance of same class is small and distinct classes are large, irrespective of imaging condition.
- Key Elements:
  - Triplet Selection
  - Triplet Loss function
  - Model Selection

Triplet Loss

$$\sum_{i}^{N}\left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha\right]_+ .$$

- Want small distance between positive class, i.e., all faces of same person should be closer.
- Tries to enforce margin between each pair of faces from 1 person to all other faces.
- Allows face of each subject to live on a manifold, with enforcing distance and thus discriminability.

Triplet Selection

- Triplet consists of anchor, positive and negative example.
- Generating all possible triplets -> not a good solution!
  - Slow training and won't contribute much to the learning.
- Sample **Hard Triplets** : triplets that violate triplet loss constraints.
- Hard Positive

$$argmax_{x^p} \left\| f(x^a) - f(x^p) \right\|_2^2$$

Hard Negative

$$argmin_{x^n} \left\| f(x^a) - f(x^n) \right\|_2^2$$

Triplet Selection

- Correct triplet selection is very crucial for convergence.
- But generate hard triplet over all training example is highly computationally inefficient.
- Offline : After n steps, calculate argmax and argmin on a subset of data.
- Online : Generate triplets within mini-batch.
- Paper implementation
  - Online triplet generation with batch size of around 1800.
  - Atleast 40 faces per subjects & additionally randomly sampled negative images.
  - Within a mini-batch, all possible positive pairs used but with hard negative.
  - Margin = 0.2

Model selection

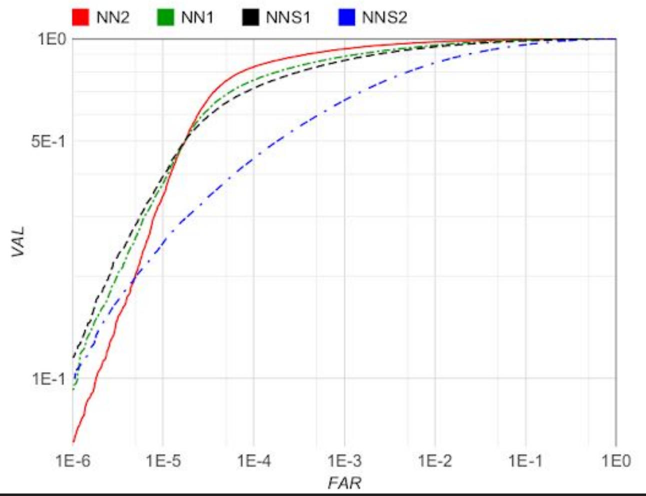
- With similar performance, implemented 2 different models for different use cases and applications.
- Servers can run inference on heavy models but for mobile applications, lightweight models required.
- Zeiler & Fergus Net
  - Heavy model
  - 140 M parameters
  - 1.6 B FLOPS / image.
- Inception Net
  - 7 M parameters (20x less)
  - 500 M FLOPS / image (5x less)

Dataset

- 8 million subjects.
- 100 million - 200 million facial images.
- 1 million test images.
- Evaluation on 4 different datasets.
- Results on face verification :
  - LFW : 99.63%
  - YouTube Faces DB : 95.12%

Experiments

- CNN Models : Computation vs Accuracy
    - Experiments with similar performing models with varying FLOPs and no. of parameters.
    - Inconclusive. Inception performed as good as Zeiler Fergus.
    - Obviously the performance will degrade after reducing the model capacity at certain point.



architecture	VAL
NN1 (Zeiler&Fergus 220×220)	87.9% ± 1.9
NN2 (Inception 224×224)	89.4% ± 1.6
NN3 (Inception 160×160)	88.3% ± 1.7
NN4 (Inception 96×96)	82.0% ± 2.3
NNS1 (mini Inception 165×165)	82.4% ± 2.4
NNS2 (tiny Inception 140×116)	51.9% ± 2.9

Table 3. **Network Architectures.** This table compares the performance of our model architectures on the hold out test set (see section 4.1). Reported is the mean validation rate VAL at 10E-3 false accept rate. Also shown is the standard error of the mean across the five test splits.

5.2. Effect of CNN Model

Experiments

- Image quality
  - Used 220x220 images for training.
  - Very low performance drop even if lower resolution images used (80x80).
- Embedding size
  - Experimented with varying embedding size upto 512. 128 works best.
  - Higher dimensional embedding require more training.
- Amount of training data
  - More data, better performance.
- SOTA result on LFW database.

Conclusion

- Provides an end-to-end model to learn face embeddings.
- Simple distance matching for various facial recognition, verification and clustering tasks.
- Better approach than traditional CNN bottleneck implementation and model concatenation.
- Robust to alignments and image specifications.
- Embeddings can be also be harmonic, i.e., embeddings generated from different models are comparable.

Thank You